KV0OOM

(aka Mike)

RETIRED = Looking for a new hobby because 11 is not enuff.

Technician + General on 18 Nov 2015

Extra on 08 Jan 2016

Not that smart, 5 years as a classroom instructor for electronics.

Many Dialects of BASIC

Dartmoth
Applesoft
Sinclair
BBC
Microsoft
Texas Instruments
QB64
BASICA
Tiny
Commodore
Liberty
GW-BASIC
PowerBasic
QuickBasic

https://en.m.wikipedia.org/basic

Many Dialects of BASIC

Dartmoth	BASICA
Applesoft	Tiny
Sinclair	Commodore
BBC	Liberty
Microsoft	**GW-BASIC**
Texas Instruments	**PowerBasic**
QB64	QuickBasic

https://en.m.wikipedia.org/basic

Legacy BASIC (GW-BASIC)

IBM-PC Interpreted BASIC, used tokens (bytes) to store programs.

A tokenized BASIC program is a text file, but is not Person readable.

This was a form of “pre-compile” to improve execution speed.

Using Programs

No issue if you copy or create a program and type it in yourself (sucks).

ARRL Antenna Compendium Vol 2 & 3

Problem occurs when you get a text file tokenized (like from the Internet) and not person readable.

Legacy Program Applications

I am primarily working with antennas:

Loading Coils

Coax Traps

Phased Array Feeds

Yagi Beam Pattern Design

Polar Pattern Plotting

Smith Chart Z Matching

Converson Options

For Windows 10…

DOSBOX = Run DOS pgms in a Window

PC-BASIC Emulator

GW-BASIC executable compiled for Windows

Old PC running DOS 6.2 or Windows 3.1

Tokenized File Conversion

If using a DOS or Win 3.1 computer, save the file (from GW-BASIC) using this command:

SAVE “FILENAME”, A

To save as a plaintext file.

Tools of Choice

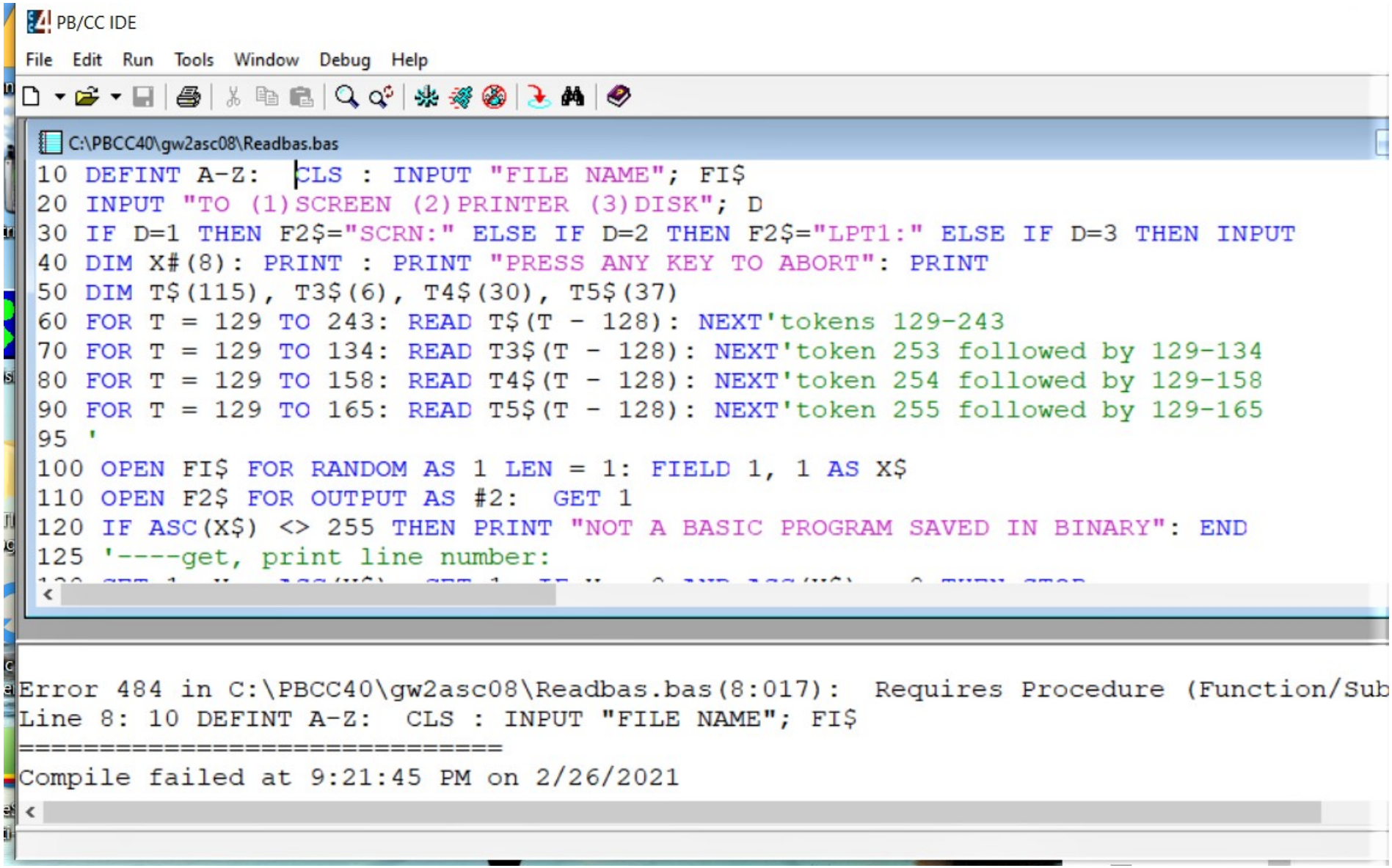
GWBASIC.EXE on a Win XP machine

PowerBasic PBCC 4.04 command line compiler.

PowerBasic started as Turbo Basic by Borlund Software.

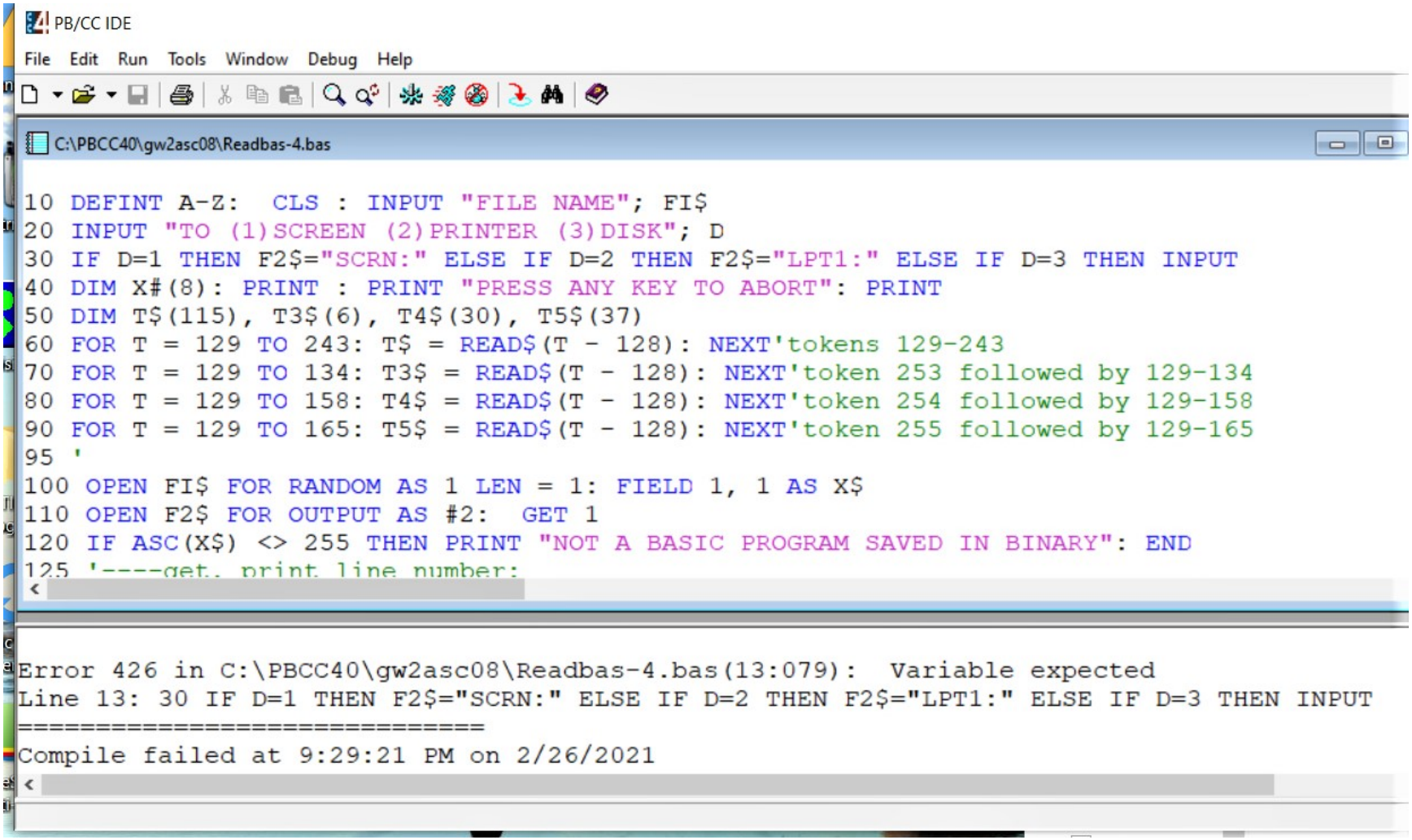
Step by Step 1

1. Compile the file and check for errors.
2. If Error 484 (requires procedure), you must add Function PBMAIN or WINMAIN at the start, and END FUNCTION at the end. This is a **WINDOWS** requirement.
3. If LPT1, PRN, or LPRINT is in the code, you need to add or modify using “XPRINT” code because these are DOS commands:


PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\Readbas.bas
10 DEFINT A-Z: CLS : INPUT "FILE NAME"; FI$
20 INPUT "TO (1)SCREEN (2)PRINTER (3)DISK"; D
30 IF D=1 THEN F2$="SCRN:" ELSE IF D=2 THEN F2$="LPT1:" ELSE IF D=3 THEN INPUT
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: READ T$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: READ T3$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: READ T4$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
Error 484 in C:\PBCC40\gw2asc08\Readbas.bas(8:017): Requires Procedure (Function/Sub
Line 8: 10 DEFINT A-Z: CLS : INPUT "FILE NAME"; FI$
==========
Compile failed at 9:21:45 PM on 2/26/2021

Step by Step 2

```
ERRCLEAR
XPRINT ATTACH DEFAULT
IF ERR = 0 AND LEN(XPRINT$)>0 THEN
‘ XPRINT “Printer working…”
‘ XPRINT FORMFEED      ‘These two lines should occur just
‘ XPRINT CLOSE          ‘ before the program exits.
ELSE
   INPUT “No Windows printer found.”; ack$
   END
END IF
```


PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\Readbas-4.bas
10 DEFINT A-Z: CLS : INPUT "FILE NAME"; FI$
20 INPUT "TO (1)SCREEN (2)PRINTER (3)DISK"; D
30 IF D=1 THEN F2$="SCRN:" ELSE IF D=2 THEN F2$="LPT1:" ELSE IF D=3 THEN INPUT
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: T$ = READ$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: T3$ = READ$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: T4$ = READ$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: T5$ = READ$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
Error 426 in C:\PBCC40\gw2asc08\Readbas-4.bas(13:079): Variable expected
Line 13: 30 IF D=1 THEN F2$="SCRN:" ELSE IF D=2 THEN F2$="LPT1:" ELSE IF D=3 THEN INPUT
==============================
Compile failed at 9:29:21 PM on 2/26/2021

Step by Step 3

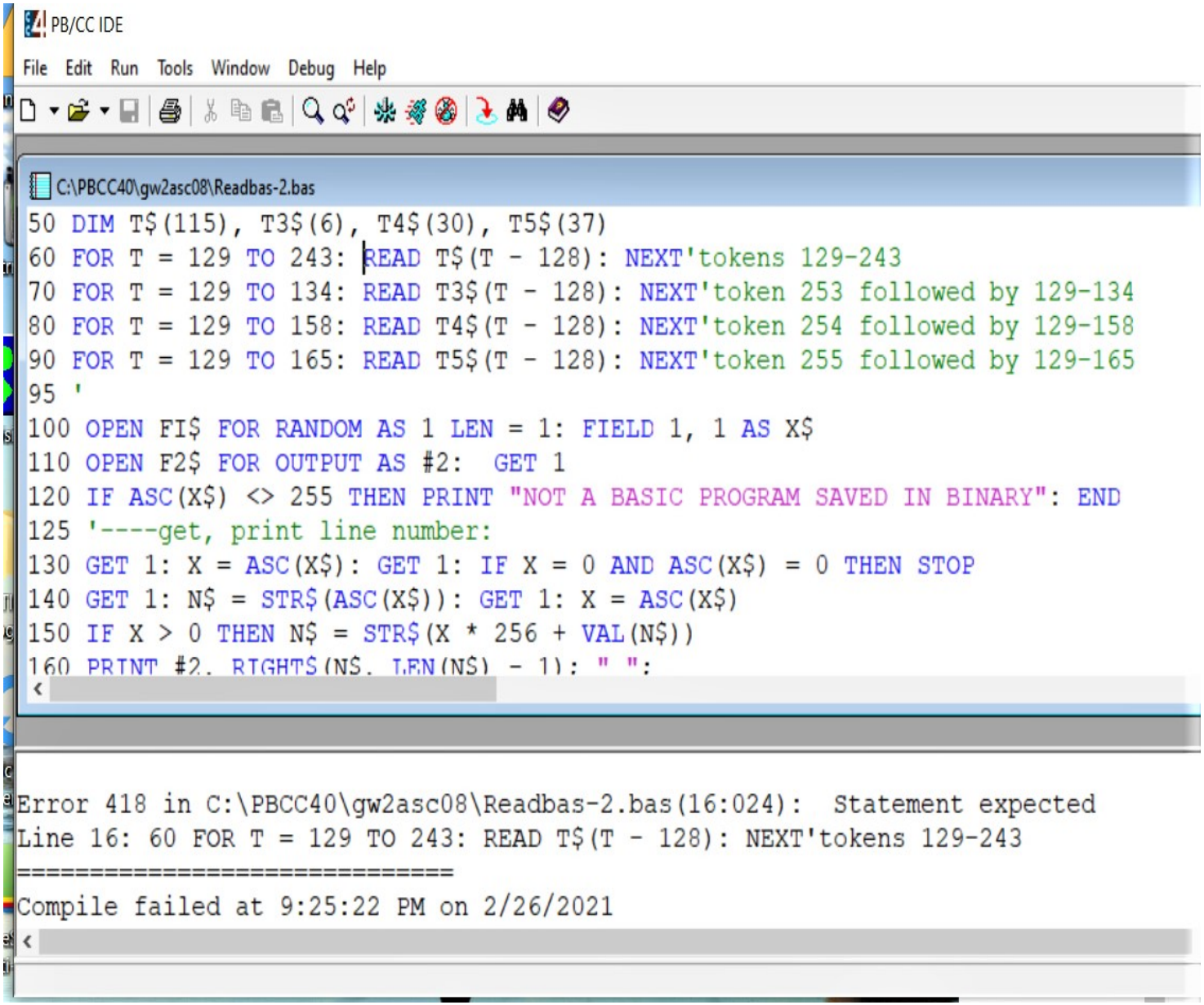
READ – DATA takes the most work, most of the time, not counting error fixing…

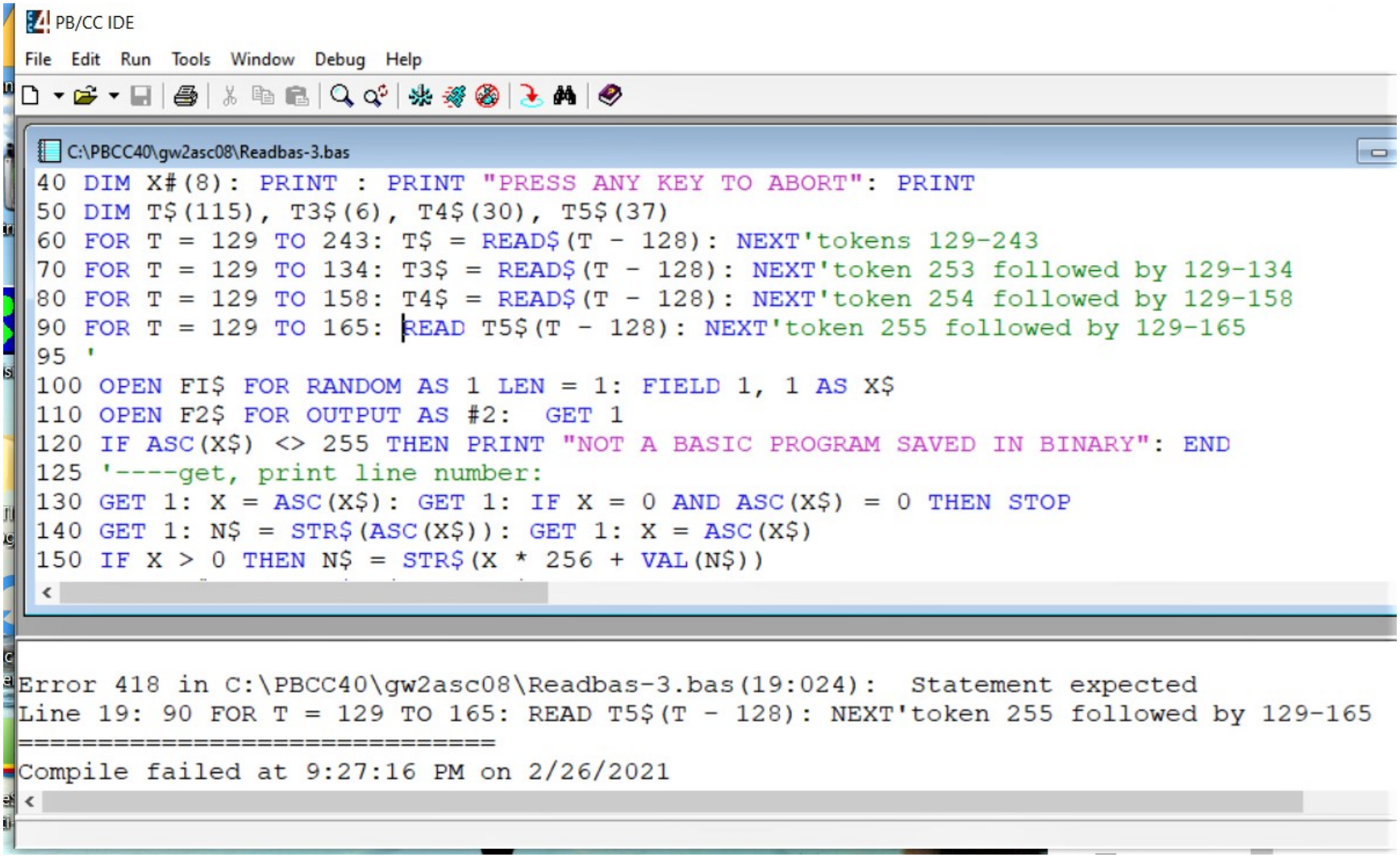
READ T5$ (x-128) ***becomes*** T5$ = READ$ (X-128)

String and Numeric data can be mixed in the same DATA line, but you have to be VERY careful forming the READs in the LOOP statement.

Eventually, you will see, “Compile succeeded at…”,

but


PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\Readbas-2.bas
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: READ T$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: READ T3$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: READ T4$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
130 GET 1: X = ASC(X$): GET 1: IF X = 0 AND ASC(X$) = 0 THEN STOP
140 GET 1: N$ = STR$(ASC(X$)): GET 1: X = ASC(X$)
150 IF X > 0 THEN N$ = STR$(X * 256 + VAL(N$))
160 PRINT #2, RIGHT$(N$, LEN(N$) - 1); " ";
Error 418 in C:\PBCC40\gw2asc08\Readbas-2.bas(16:024): Statement expected
Line 16: 60 FOR T = 129 TO 243: READ T$(T - 128): NEXT'tokens 129-243
==========
Compile failed at 9:25:22 PM on 2/26/2021


PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\Readbas-3.bas
40 DIM X#(8): PRINT : PRINT "PRESS ANY KEY TO ABORT": PRINT
50 DIM T$(115), T3$(6), T4$(30), T5$(37)
60 FOR T = 129 TO 243: T$ = READ$(T - 128): NEXT'tokens 129-243
70 FOR T = 129 TO 134: T3$ = READ$(T - 128): NEXT'token 253 followed by 129-134
80 FOR T = 129 TO 158: T4$ = READ$(T - 128): NEXT'token 254 followed by 129-158
90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
95 '
100 OPEN FI$ FOR RANDOM AS 1 LEN = 1: FIELD 1, 1 AS X$
110 OPEN F2$ FOR OUTPUT AS #2: GET 1
120 IF ASC(X$) <> 255 THEN PRINT "NOT A BASIC PROGRAM SAVED IN BINARY": END
125 '----get, print line number:
130 GET 1: X = ASC(X$): GET 1: IF X = 0 AND ASC(X$) = 0 THEN STOP
140 GET 1: N$ = STR$(ASC(X$)): GET 1: X = ASC(X$)
150 IF X > 0 THEN N$ = STR$(X * 256 + VAL(N$))
Error 418 in C:\PBCC40\gw2asc08\Readbas-3.bas(19:024): Statement expected
Line 19: 90 FOR T = 129 TO 165: READ T5$(T - 128): NEXT'token 255 followed by 129-165
==========
Compile failed at 9:27:16 PM on 2/26/2021

Step by Step 4

YOU ARE NOT DONE.

You still have to run the program.

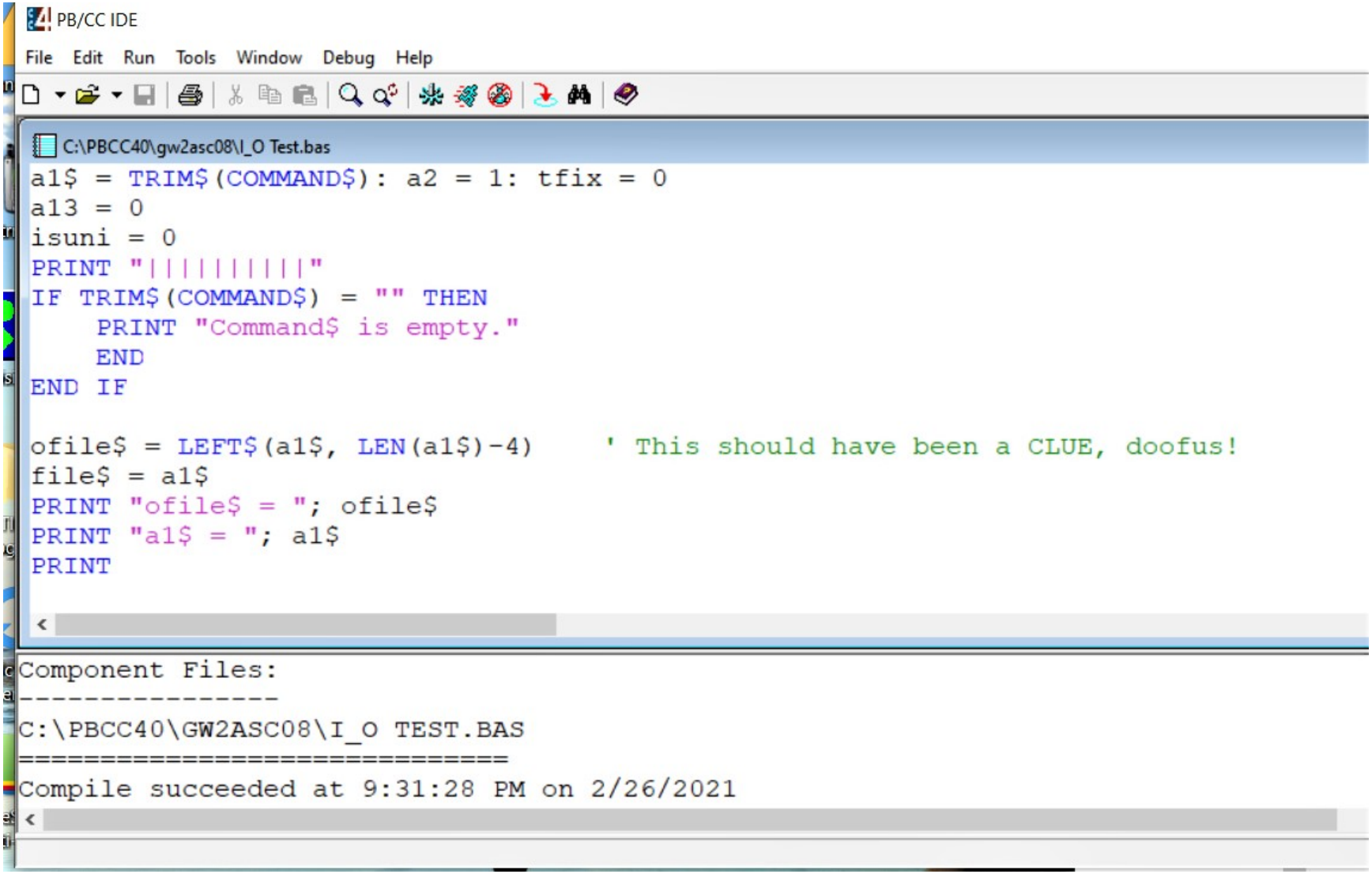
Double click or drag&drop on the EXE, it is a Windows program now.

Unfortunately…

The program filename parsing routines assumed the original 8.3 file name convention, which appear relevant when using file I/O operations.

Also, you may encounter a program where the programmer is showing off his newly learned keywords. For example, he/she may use MID$() where LEFT$() was more appropriate. Decoding what they intended can sometimes be very interesting.

This is where familiarity with BASIC comes in handy.


PB/CC IDE
File Edit Run Tools Window Debug Help
C:\PBCC40\gw2asc08\I_O Test.bas
a1$ = TRIM$(COMMAND$): a2 = 1: tfix = 0
a13 = 0
isuni = 0
PRINT "||||||||||"
IF TRIM$(COMMAND$) = "" THEN
PRINT "Command$ is empty."
END
END IF
ofile$ = LEFT$(a1$, LEN(a1$)-4) ' This should have been a CLUE, doofus!
file$ = a1$
PRINT "ofile$ = "; ofile$
PRINT "a1$ = "; a1$
PRINT
Component Files:
C:\PBCC40\GW2ASC08\I_O TEST.BAS
Compile succeeded at 9:31:28 PM on 2/26/2021

Quirks 1

Sometimes code does not work as expected, or desired. You have to decide if the effort to “clean it up” is worth it. Should you cut your losses and run?

For me, it becomes personal. I like to think I am smarter than a glorified calculator.

Quirks 2

For example, I eventually discovered COMMAND$ returns the file path with quotes (I suspect because it contained a blank space) when using Windows drag & drop (drag the .BAS and drop it on the new EXE). I missed 2 clues with LEN and PRINT, and it cost me 15-16 hours of debugging.

Cut Your Losses and Run, Away

The program was to convert tokenized BASIC to ASCII text. Once I got thru the file I/O issues, I learned the program was reporting token errors.

I had already created an ASCII copy of the program. I knew the .BAS program was error free (token-wise) and knew what the output should be.

Project Tabled

I may return to this project, but I know it will be a lot of work.

1. Locating the correct tokenization table for this version of BASIC.
2. Figuring out why the original programmer's code does not work.
3. Outsmarted is a coding issue. Aggravation is a programmer's code issue.

Pretend it Worked

We added PBMAIN or WINMAIN. Next are those pesky “LPRINT” statements.

LPRINT is a DOS function and we are in a Windows box. This means there will be some code edits required because DOS is deeply linked to the machine hardware. Windows is not.

LPRINT becomes XPRINT

First, we must tell the program how to connect to the Windows printer:

```
ERRCLEAR
XPRINT ATTACH DEFAULT
```

Then test for success…

```
IF ERR = 0 AND LEN (XPRINT$)>0 THEN
‘   XPRINT “Printer working…”
ELSE
   INPUT “No Windows printer found.”; ack$
   END
END IF
```

Editing for Printer Output

Now we are ready for the paper copy statements. If you change LPRINT or PRINT keywords to XPRINT, no other editing is normally needed.

I prefer to convert the LPRINTs to XPRINTs, and make a second copy of the PRINT statements. I turn those into XPRINTs and edit for a pretty printout.

Reading DATA Revisited

READ$ is a TEXT command. It is best to keep numeric DATA separate from string DATA. However, it is possible to read the data in groups of variables in each pass of the FOR-NEXT loop.

Once I got away from TRS-80 Level II BASIC, I rarely used DATA statements, so my experience here is limited.

The Last Code Part

These should be the last 3 lines of code if XPRINTing:

XPRINT FORMFEED

XPRINT CLOSE

END FUNCTION

Be sure to CLOSE the printer before ending the program.

GWBASIC.EXE

If you need a copy of GW-BASIC, you can download it from here:

https://hwiegman.home.xs4all.nl/gwbasit.html

This is where I got my copy because Win XP did not include it.

More Links

Detokenizers (Microsoft BASICs) Info

http://edmond.orignac.pagesperso-orange.fr/detokenizer.html

For Syntax:

http://www.antonis.de/qbebooks/gwbasicman

HEX editors: (UltraEdit and HdX)

http://ultraedit.freedownloadscenter.com/windows/free

https://mh-nexus.de/en/hxd/

HEX Editor?

The HEX editor is a special tool for looking at files other than with a text editor. GW-BASIC tokenized files start with “**FF**” hex as the first byte in the file. If the file is “protected”, the first byte will be “**FE**” hex.

And no…, you cannot change the first byte to something else so you can unlock the file.

PEEK – POKE

If you see either of these keywords in the code, FORGET IT. These commands are hardware specific and DOS connected. Windows does not allow random peeking and poking memory. You will get a memory violation or other dramatic end to your program.

If you can’t resist the need for speed, then learn Assembly Language for x86 processors. PowerBasic allows inline assembly embedded in the BASIC source code.

KV0OOM

This project was initiated by obtaining some files authored by A. C. Buxton, W8NX. If you delve into coax chokes and traps, you will see this name a lot.

kv0oom, Amateur Extra.
kv0oom.extra@gmail.com
Old, but not an Old Timer.
Google is your best friend.
RETIRED and hobby poor!

END

Questions?

Software demo is next…